



**Министерство науки и высшего образования Российской
Федерации
Федеральное государственное бюджетное образовательное
учреждение
высшего образования
«Московский государственный технический
университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ «ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ»

**КАФЕДРА «ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И
ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1 ПО
ДИСЦИПЛИНЕ:**

ТИПЫ И СТРУКТУРЫ ДАННЫХ

**ТЕМА: "ДЛИННАЯ АРИФМЕТИКА": ОБРАБОТКА
БОЛЬШИХ ЧИСЕЛ**

Студент Поздышев А. В.

Группа ИУ7-31Б

Название предприятия НУК ИУ МГТУ им. Н. Э. Баумана

Студент _____ Поздышев А. В.

Преподаватель _____ Силантьева А. В.

2024

Оглавление	
ОПИСАНИЕ УСЛОВИЯ ЗАДАЧИ.....	2
ОПИСАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ.....	2
ОПИСАНИЕ ВНУТРЕННИХ СТРУКТУР ДАННЫХ	3
ФУНКЦИИ ПРОГРАММЫ.....	3
ОПИСАНИЕ АЛГОРИТМА.....	4
НАБОР ТЕСТОВ	5
ВЫВОДЫ	7
ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ	7

ОПИСАНИЕ УСЛОВИЯ ЗАДАЧИ

Смоделировать операцию деления действительного числа на действительное число в форме $[\pm]m[.]n[E[\pm]K]$, где суммарная длина мантиссы ($m+n$) - до 40 значащих цифр, а величина порядка K - до 5 цифр. Результат выдать в форме $\pm 0.m1 E \pm K1$, где $m1$ – до 40 значащих цифр, а $K1$ - до 5 цифр.

ОПИСАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ

Входные данные

Две строки, содержащие действительные числа в экспоненциальной или обычной форме.

Действительное число – строка вида $\langle [\pm]m[.]n[E[\pm]K] \rangle$, где длина мантиссы ($m+n$) – до 40 значащих цифр, а K – до 5 цифр.

Выходные данные

Строка, содержащая в себе результат деления двух введенных действительных чисел.

Результат деления двух действительных чисел представляется в виде $\langle \pm 0.m1 E \pm K1 \rangle$, где $m1$ – мантисса до 40 значащих цифр, а $K1$ - до 5 цифр.

Описание задачи, реализуемой программой

Задача программы – осуществление деления одного действительного числа на другое действительное число.

Способ обращения к программе

Ввод и вывод всех данных осуществляется через консоль.

Описание возможных аварийных ситуаций и ошибок пользователя

Аварийные ситуации:

- Пустое поле ввода (ожидание ввода пользователя);
- Переполнения порядка при делении;
- Деление на ноль;

Ошибки пользователя:

- Некорректный ввод: превышение допустимой длины мантиссы ($m+n$);
- Некорректный ввод: превышение допустимой длины значения K ;
- Некорректный ввод: посторонние символы в записи числа;
- Некорректный ввод: запись числа, не подходящая маске $\langle [\pm]m[.]n[E[\pm]K] \rangle$;

ОПИСАНИЕ ВНУТРЕННИХ СТРУКТУР ДАННЫХ

Введенные действительные числа в программе представляются в структурном типе `big_num`.

Листинг структуры `big_num`:

```
#define LEN_MANTISA 41
struct big_num
{
    int massive[LEN_MANTISA];

    int point_pos;

    size_t len;

    char sign;

    int power;
};
```

1. `massive` – массив целых чисел, элементами которого являются цифры мантиссы введенного действительного числа.
2. `point_pos` – индекс позиции точки в действительном числе.
3. `len` – длина мантиссы введенного числа.
4. `sign` – знак введенного действительного числа.
5. `power` – значение порядка введенного действительного числа.

ФУНКЦИИ ПРОГРАММЫ

`Complex_devide`

Заголовок:

```
int complex_devide(struct big_num *num_1, struct big_num *num_2, struct big_num *result);
```

Функция осуществляет деление действительных чисел типа `big_num`. Во время ее выполнения она вызывает такие функции как: `simple_multy` и `to_sub`. Функция принимает указатели типа `big_num` на делимое `num_1`, делитель `num_2`, частное `result`. Возвращает код ошибки выполнения операции.

`Simple_multy`

Заголовок:

```
int simple_multy(struct big_num *num, int multy);
```

Функция осуществляет умножение действительного числа типа `big_num` на однозначное типа `int`. Функция принимает указатель типа `big_num` на число

num, однозначное число multy. Возвращает код ошибки выполнения операции.

To_sub

Заголовок:

```
int to_sub(struct big_num *num_1, struct big_num *num_2);
```

Функция осуществляет вычитание целых чисел типа big_num. Функция принимает указатели типа big_num на уменьшаемое num_1, вычитаемое num_2. Разность записывается в num_1. Возвращает код ошибки выполнения операции.

Take_num

Заголовок:

```
int take_num(char const *buffer, struct big_num *num);
```

Функция считывает из буфера число структурного типа big_num. Функция принимает буфер buffer, указатель типа big_num на число num. Возвращает код ошибки, если строка не подходит маске $[\pm]n[.]m[E[\pm]k]$, где $(n + m) = 40$ до 40 цифр, k до 5 цифр.

Print_num

Заголовок:

```
void print_num(struct big_num num);
```

Функция выводит число структурного типа big_num на экран в виде $\pm 0.mEk$, где m до 40 цифр. Функция принимает число num типа big_num.

Print_err

Заголовок:

```
void print_err(int const rc);
```

Функция выводит сообщение об ошибке. Функция принимает код ошибки rc.

ОПИСАНИЕ АЛГОРИТМА

Программа предлагает пользователю ввести два действительных числа. После введения строк программа считывает их и проверяет каждую из них на валидность. Если строки не корректны, то выводится сообщение об ошибке ввода.

Во время выполнения алгоритма в начале оба действительных числа приводятся к целому ввиду за счет изменения их порядков и смещения позиции точки. На втором шаге находится неполное делимое большее делителя за счет сравнивания старших разрядов неполного делимого и

Результат деления двух чисел выводится в виде $\langle \pm 0.m1E\pm K1 \rangle$, где m1 до 40 цифр, а K1 до 5 цифр.

№	Название	Число №1	Число №2	Результат
1	Некорректный ввод	abc	-	ERR_IO: Ошибка ввода.
2	Некорректный ввод	+abc.defe+10	-	ERR_IO: Ошибка ввода.
3	Некорректный ввод	e+10	-	ERR_IO: Ошибка ввода.
4	Кол-во цифр в мантиссе больше 40	99999999999999999999. 999999999999999999e1	-	ERR_IO: Ошибка ввода.
5	Кол-во цифр в порядке больше 5	123.123e123456	-	ERR_IO: Ошибка ввода.

	сторон у			
1 6	Округ ление в меньш ую сторон у	1	3	+0.3333333333333333 333333333333333333 333e0
1 7	Делен ие чисел с разны ми знакам и	+15	-2	-0.75e1
1 8	Перен ос округл ения на старш ие разряд ы	99999999999999999999 999999999999999999	2	+0.5e40

ВЫВОДЫ

Для представления больших действительных чисел в памяти компьютера очень удобно использовать такой тип данных как структура. С помощью структуры можно разделить число на его компоненты, что значительно упрощает хранение и дальнейшую работу с ним.

ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Каков возможный диапазон чисел, представляемых в ПК?

Диапазон чисел зависит от размера области памяти, выделенной компьютером. Например, тип `int` занимает 4 байт – диапазон от -2 147 483 648 до 2 147 483 647 для 64-разрядных ПК.

2. Какова возможная точность представления чисел, чем она определяется?

Вещественные числа хранятся в представлении с плавающей точкой в виде $X = M * n^p$, где M – мантисса в виде дроби. Точность числа зависит от максимально возможной длины этой мантиссы, которая зависит от области выделенной памяти (кол-во бит). При выходе за длину мантиссы происходит

округление. Для n -битного целого числа максимальная представленность (максимальное значение) определяется формулой: $2^n - 1$.

Для 64 разрядов это $2^{64} - 1 = 18\,446\,744\,073\,709\,551\,615$.

3. Какие стандартные операции возможны над числами?

Стандартные операции над числами – сложение, вычитание, умножение, деление.

4. Какой тип данных может выбрать программист, если обрабатываемые числа превышают возможный диапазон представления чисел в ПК?

Для представления больших чисел в памяти компьютера можно использовать структуры, в которой будет храниться мантисса, порядок, знак, цифры числа и другое.

5. Как можно осуществить операции над числами, выходящими за рамки машинного представления?

Операции над большими числами выполняются путем последовательного выполнения арифметических операций над их цифрами, которые хранятся массиве.